HPL Project Instructions

1. Initial Setup:

a. **Choose Cloud Provider**: Start by selecting a cloud provider like AWS, Azure, or Google Cloud. Sign in to your chosen provider's dashboard.

b. **Create Virtual Machines**: Create multiple virtual machine (VM) instances in your cloud provider's infrastructure. You'll need several VMs for this project, and they should be distributed across different regions and network configurations as specified in your objectives.

c. **Networking**: Ensure that networking configurations, such as Virtual Private Cloud (VPC), Subnets, and Security Groups, are correctly set up to allow communication between instances when needed. For internal network communication, instances should be in the same VPC or network.

2. HPL Benchmark Setup:

a. **Choose a Reference Instance**: Pick one of your VM instances as the reference instance where you will download and compile the HPL benchmark.

b. **Download and Compile HPL**: SSH into the reference instance and download the HPL benchmark. Follow the instructions provided with HPL to compile it on your instance.

c. **Problem Size Optimization**: Run HPL with different problem sizes on the reference instance. This helps you find the optimal problem size that yields the highest FLOPS (Floating Point Operations Per Second) performance. Record the FLOPS achieved for each problem size.

3. MPI Installation and Testing:

a. **Install MPI**: Install the Message Passing Interface (MPI) implementation of your choice (e.g., MPICH or OpenMPI) on all VM instances. Make sure to install the same MPI version and configure it appropriately.

b. **MPI Testing**: Create a simple MPI test program (e.g., a "Hello, World!" program) and run it to verify that VM instances can communicate with each other through MPI. Ensure that firewall rules or security groups allow MPI communication.

4. Running HPL Benchmark:

a. **Prepare HPL Configuration**: Create an HPL configuration file specifying the problem size determined in step 2. Configure other parameters such as block size, grid size, and the number of CPU cores.

b. **Scenario 1: Public Network in Same Region**: Run HPL on instances in the same region but accessed through the public network. Record FLOPS and other relevant metrics.

c. **Scenario 2: Internal Network**: Run HPL on instances accessed through the internal network. Record FLOPS and other metrics.

d. **Scenario 3: Different Regions via Public Network**: Run HPL on instances in different regions, accessed through the public network. Record FLOPS and metrics.

e. **Additional Scenarios**: As mentioned in your objectives, you can explore other scenarios that you think might yield interesting results.

5. Performance Metrics and Analysis:

a. **FLOPS Calculation**: Calculate the FLOPS achieved in each scenario using the execution time and the Linpack benchmark's problem size. FLOPS = (Problem Size in Floating Point Operations) / (Execution Time in Seconds)

b. **iPerf Bandwidth Test**: Use iPerf to measure the highest achievable bandwidth between VM instances in different network scenarios. Record the results.

c. **Bottleneck Identification**: Analyze the results to identify bottlenecks. Compare FLOPS and bandwidth across different scenarios to pinpoint performance limitations. Possible bottlenecks could include network latency, bandwidth constraints, or differences in instance types.

6. Conclusion and Reporting:

a. **Conclusion**: Summarize your findings and discuss how instance location and network interfaces affect cluster performance.

b. **Recommendations**: Provide recommendations based on your findings, such as network optimizations or instance type choices for specific use cases.

c. **Visualizations**: Create visualizations, charts, and graphs to illustrate your results. Visuals can make your findings more accessible.

d. **Documentation**: Document the entire process, including setup, configuration, benchmarking commands, and results. This documentation will be valuable for replicating the project.

e. **Sharing Results**: Make your project report and findings available for further research or reference, especially if you discover novel insights or optimizations.

This detailed breakdown should guide you through each step of the project, from initial setup to the analysis of results. Remember to document your progress and findings thoroughly, as this will be essential for drawing meaningful conclusions and sharing your insights.

how to download and compile the HPL benchmark on AWS VM

To download and compile the HPL (High-Performance Linpack) benchmark on an AWS Virtual Machine (VM), you'll need to follow these general steps:

1. **Launch an AWS VM**:
   - Log in to your AWS Management Console.
   - Create a new EC2 instance with your desired specifications (e.g., instance type, region, and operating system). Ensure that you have the necessary permissions and key pairs to access the instance.
2. **Connect to Your AWS VM**:
   - Once your AWS VM is running, connect to it using SSH. You can use the private key associated with your instance to log in.
3. **Install Required Dependencies**:
   - HPL has some dependencies that need to be installed. The specific commands may vary depending on your VM's operating system. For example, on a typical Linux system, you might use `yum` or `apt-get`:

csharp Copy code

```
# For Ubuntu/Debian
sudo apt-get update
sudo apt-get install build-essential openmpi-bin libopen

# For CentOS/RHEL
sudo yum groupinstall "Development Tools"
sudo yum install openmpi openmpi-devel
```

Re

4. **Download and Extract HPL**:
   - You can download the HPL source code from the official website or a trusted source. Here's an example of how to download and extract it:

```bash
wget http://www.netlib.org/benchmark/hpl/hpl-2.3.tar.gz
tar -xzvf hpl-2.3.tar.gz
cd hpl-2.3
```

5. **Configure HPL**:
   - Before compiling, you need to create a configuration file for HPL. You can use the included setup script for this purpose:

```arduino
cp setup/Make.Linux setup/Make.Linux
```

   Edit the `Make.Linux` file to customize the configuration based on your system's specifications (e.g., processor type, compiler, and BLAS library). You can use a text editor like `nano` or `vim` to edit the file.

6. **Compile HPL**:
   - After configuring HPL, you can compile it using the `make` command:

```bash
make arch=Linux
```

7. **Run the HPL Benchmark**:
   - Once the compilation is complete, you can run the HPL benchmark. To do this, you need to create a hostfile specifying the hosts in your cluster. Then, you can use the `mpirun` command to start the benchmark:

```php
mpirun -np <number_of_processes> -hostfile <hostfile> ./
```

   Replace `<number_of_processes>` with the desired number of MPI processes and `<hostfile>` with the path to your hostfile.

8. **Monitor and Analyze Results**:
   - Monitor the benchmark as it runs and collect the results. HPL will provide performance metrics that can be used to evaluate the system's performance.